
Langage de définition de données

L'instruction SQL CREATE DATABASE :

L'instruction CREATE DATABASE est utilisée pour créer une nouvelle base de données SQL.

```
CREATE DATABASE databasename;
```

Exemple :

```
CREATE DATABASE testDB;
```

L'instruction SQL DROP DATABASE

L'instruction DROP DATABASE est utilisée pour supprimer une base de données SQL existante.

Syntaxe :

```
DROP DATABASE databasename;
```

Exemple

```
DROP DATABASE testDB;
```

Création des tables - CREATE TABLE

Syntaxe :

```
1 CREATE TABLE nom_table(  
2   column1 type_donnees [contraintes],  
3   column2 type_donnees [contraintes],  
4   column3 type_donnees [contraintes],  
5   .....  
6   columnN type_donnees [contraintes],  
7   PRIMARY KEY( une ou plusieurs colonnes )  
8);
```

Exemple :

```
CREATE TABLE Employes(  
  Id INT NOT NULL,  
  Nom VARCHAR (20) NOT NULL,  
  Age INT NOT NULL,  
  Salaire DECIMAL (18, 2),  
  PRIMARY KEY (Id)  
);
```

SQL ALTER TABLE

La commande ALTER TABLE en SQL permet de modifier une table existante. Idéal pour ajouter une colonne, supprimer une colonne ou modifier une colonne existante, par exemple pour changer le type.

Syntaxe de base :

```
ALTER TABLE nom_table  
instruction
```

SQL ALTER TABLE

Ajouter une colonne :

Syntaxe :

```
ALTER TABLE nom_table  
ADD nom_colonne type_donnees
```

Exemple

```
ALTER TABLE utilisateur  
ADD adresse_rue VARCHAR(255)
```

SQL ALTER TABLE

Supprimer une colonne

```
ALTER TABLE nom_table  
DROP nom_colonne
```

OU

```
ALTER TABLE nom_table  
DROP COLUMN nom_colonne
```

SQL ALTER TABLE

Modifier une colonne

```
ALTER TABLE nom_table
```

```
MODIFY nom_colonne type_donnees
```

Renommer une colonne :

```
ALTER TABLE nom_table
```

```
CHANGE colonne_ancien_nom colonne_nouveau_nom type_donnees
```

SQL ALTER TABLE

Ajouter une contrainte :

ADD peut également être utilisé pour créer une contrainte sur les colonnes de la table

Syntaxe :

```
ALTER TABLE nom_table
```

```
    ADD CONSTRAINT nom_contrainte contrainte
```

Exemple :

La requête suivante, ajoutera une contrainte CHECK à la colonne Age.

```
ALTER TABLE Employes
```

```
    ADD CONSTRAINT ageContrainte CHECK(Age >= 18);
```

Suppression des tables - DROP TABLE

La commande DROP TABLE en SQL permet de supprimer définitivement une table d'une base de données. Cela supprime en même temps les éventuels index, trigger, contraintes et permissions associées à cette table. Attention : il faut utiliser cette commande avec attention car une fois supprimée, les données sont perdues. Avant de l'utiliser sur une base importante il peut être judicieux d'effectuer un backup (une sauvegarde) pour éviter les mauvaises surprises.

Syntaxe :

DROP TABLE nom_table

A savoir : s'il y a une dépendance avec une autre table, il est recommandé de les supprimer avant de supprimer la table. C'est le cas par exemple s'il y a des clés étrangères.

Insertion des enregistrements - INSERT INTO

L'insertion de données dans une table s'effectue à l'aide de la commande INSERT INTO. Cette commande permet au choix d'inclure une seule ligne à la base existante ou plusieurs lignes d'un coup.

Insérer une ligne en spécifiant toutes les colonnes :

```
INSERT INTO table VALUES ('valeur 1', 'valeur 2', ...)
```

|

Insertion des enregistrements - INSERT INTO

Insertion de plusieurs lignes à la fois

```
INSERT INTO client (prenom, nom, ville, age)
```

```
VALUES
```

```
('Rébecca', 'Armand', 'Saint-Didier-des-Bois', 24),
```

```
('Aimée', 'Hebert', 'Marigny-le-Châtel', 36),
```

```
('Marielle', 'Ribeiro', 'Maillères', 27),
```

```
('Hilaire', 'Savary', 'Conie-Molitard', 58);
```

Modification des enregistrements - UPDATE

La commande UPDATE permet d'effectuer des modifications sur des lignes existantes. Très souvent cette commande est utilisée avec WHERE pour spécifier sur quelles lignes doivent porter la ou les modifications.

Syntaxe :

UPDATE table

SET nom_colonne_1 = 'nouvelle valeur'

WHERE condition

Exemple :

```
UPDATE client SET pays = 'FRANCE' where id=2
```

Suppression des enregistrements - DELETE

La commande DELETE en SQL permet de supprimer des lignes dans une table. En utilisant cette commande associé à WHERE il est possible de sélectionner les lignes concernées qui seront supprimées.

Syntaxe :

```
DELETE FROM `table` WHERE condition  
DELETE FROM `utilisateur` WHERE `id` = 1
```

Supprimer plusieurs lignes

```
DELETE FROM `utilisateur`  
WHERE `date_inscription` < '2012-04-10'
```

Supprimer toutes les données :

```
DELETE FROM `utilisateur`
```